

INTELIGENCIA ARTIFICIAL

Producto Integrador de Aprendizaje- Planificador

Daniel Gerardo Ortiz Garza

1901157

Alejandro Rodarte Pérez

1925234

Descripción del problema

- En este problema elaboramos un **planificador** que nos da un proceso sobre como cocinar **milanesa de pollo** utilizando **PDDL** [Planning Domain Definition Language]
- Debido a ciertas circunstancias en la instalacion de docker optamos por utilizar la pagina web:
- **<http://editor.planning.domains>**

Utilizamos el **problema**
01 del **dominio de**
Películas para guiarnos
en la realización del
proyecto

PROBLEMA

“Init” nos indica el estado inicial de el problema

En este problema tenemos como estado inicial 12 objetos enumerados en la imagen de la izquierda y una estufa encendida.

La meta de este problema es lograr el pollo cocido, que el pollo este en el recipiente y que la estufa este prendida

```
(define (problem hacer-pollo)
  (:domain hacer-pollito)
  (:objects p5 p4 p3 p2 p1 m2 m1 r3 r2 r1 s2 s1 )

  (:init (pollo p5)
          (pollo p4)
          (pollo p3)
          (pollo p2)
          (pollo p1)
          (mantequilla m2)
          (mantequilla m1)
          (recipiente r3)
          (recipiente r2)
          (recipiente r1)
          (sarten s2)
          (sarten s1)
          (estufa-on))

  (:goal (and (pollo-cocido)
              (polloEnRecipiente)
              (estufa-on)))
)
```

DOMINIO- PREDICADOS

Los predicados son las propiedades de los objetos o pueden ser Verdaderos/falsos

En este caso tenemos 13 predicados de los cuales 4 son nuestra “materia prima”

```
(:predicates (mantequilla-derretida)
              (tener-mantequilla)
              (estufa-on)
              (pollo-cocido)
              (tener-pollo)
              (tener-sarten)
              (sarten-caliente)
              (polloEnRecipiente)
              (tener-recipiente)
              (pollo ?x)
              (mantequilla ?x)
              (recipiente ?x)
              (sarten ?x))
```

DOMINIO-ACCIONES

Las acciones son formas de cambiar el estado.

Las acciones `get` en este caso son las que nos hacen obtener el objeto y todas tienen de precondicion `?x` es decir solo entrara la acción en caso de tener el producto en “stock”.

Las acciones `put` son las que nos permiten poner dicho objeto, ya sea en el recipiente o en la estufa.

Las acciones que logran nuestra meta es La acción `put pollo` que nos genera el efecto de `pollo-cocido` y la de servir pollo que ocasiona el efecto `PolloEnRecipiente`

```
(:action get-sarten
  :parameters (?x)
  :precondition (and (sarten ?x))
  :effect (tener-sarten)
)

(:action put-sarten
  :parameters ()
  :precondition (and (estufa-on) (tener-sarten))
  :effect (sarten-caliente)
)

(:action get-mantequilla
  :parameters (?x)
  :precondition (mantequilla ?x)
  :effect (tener-mantequilla)
)

(:action put-mantequilla
  :parameters ()
  :precondition (and (tener-mantequilla) (sarten-caliente))
  :effect (and (mantequilla-derretida))
)

(:action get-pollo
  :parameters (?x)
  :precondition (pollo ?x)
  :effect (tener-pollo)
)

(:action put-pollo
  :parameters ()
  :precondition (and (mantequilla-derretida) (tener-pollo))
  :effect (pollo-cocido)
)

(:action get-recipiente
  :parameters (?x)
  :precondition (and (recipiente ?x))
  :effect (and (tener-recipiente))
)

(:action servir-pollo
  :parameters ()
  :precondition (and (tener-recipiente) (pollo-cocido))
  :effect (and (polloEnRecipiente))
)
```

Plan

Found Plan (output)

(get-pollo p5)

(get-mantequilla m2)

(get-sarten s2)

(put-sarten)

(put-mantequilla)

(put-pollo)

(get- recipiente r3)

(servir-pollo)

```
(:action get-pollo
:parameters (p5)
:precondition
  (pollo p5)
:effect
  (tener-pollo)
)
```

<https://github.com/DanielOrtiz09/Inteligencia-Artificial>